

Panduan Scrum

Panduan Definitif untuk Scrum:
Aturan Main

November 2017



Jeff Sutherland

Ken Schwaber

Dikembangkan dan dipertahankan oleh pencipta Scrum: Ken Schwaber dan Jeff Sutherland

BAHASA INDONESIA

Daftar Isi

Tujuan dari Panduan Scrum.....	3
Definisi dari Scrum.....	3
Penggunaan Scrum	3
Teori Scrum.....	4
Tata Nilai Scrum.....	5
Scrum Team	5
<i>Product Owner</i>	6
<i>Development Team</i>	6
<i>Scrum Master</i>	7
Acara-acara Scrum.....	8
<i>Sprint</i>	8
<i>Sprint Planning</i>	9
<i>Daily Scrum</i>	11
<i>Sprint Review</i>	11
<i>Sprint Retrospective</i>	12
Artefak-artefak Scrum	13
<i>Product Backlog</i>	13
<i>Sprint Backlog</i>	14
<i>Increment</i>	15
Transparansi Artefak	15
Definisi “Selesai”	16
Catatan Penutup.....	17
Ucapan Terima Kasih	17
Pihak terkait.....	17
Sejarah.....	17
Penghargaan untuk penerjemah.....	17
Perubahan dari versi 2016 ke versi 2017	18

Tujuan dari Panduan Scrum

Scrum adalah sebuah kerangka kerja untuk mengembangkan, menghantarkan dan mengelola produk yang kompleks. Panduan ini berisi definisi Scrum yang terdiri dari peran-peran, acara-acara, artefak-artefak, dan aturan-aturan yang menyatukan kesemuanya. Ken Schwaber dan Jeff Sutherland mengembangkan Scrum; Panduan Scrum ditulis dan disebarluaskan oleh mereka. Mereka berdua yang tetap mempertahankan Panduan Scrum.

Definisi dari Scrum

Scrum (kb): Sebuah kerangka kerja dimana orang-orang dapat mengatasi masalah kompleks adaptif, dimana pada saat bersamaan mereka juga menghantarkan produk dengan nilai setinggi mungkin secara produktif dan kreatif.

Scrum bersifat :

- Ringan
- Sederhana untuk dipahami
- Sulit untuk dikuasai

Scrum adalah kerangka kerja proses yang telah digunakan untuk mengelola pengembangan produk kompleks sejak awal tahun 1990-an. Scrum bukanlah sebuah proses, teknik, ataupun metodologi. Akan tetapi Scrum adalah sebuah kerangka kerja dimana anda dapat menggunakan bermacam proses dan teknik di dalamnya. Scrum mengekspos ketidak-efektifan dari manajemen produk dan teknik kerja anda, sehingga anda dapat secara terus-menerus meningkatkan kinerja produk, tim, dan lingkungan kerja anda.

Kerangka kerja Scrum terdiri dari *Scrum Team* dan peran-peran, acara-acara, artefak-artefak dan aturan-aturan terkait. Setiap komponen di dalam kerangka kerja ini memiliki tujuan tertentu dan sangat penting bagi keberhasilan penggunaan Scrum.

Aturan Scrum mengikat peran-peran, acara-acara, dan artefak-artefak, serta menjaga hubungan dan interaksi antar komponen tersebut. Aturan Scrum dijelaskan di sepanjang dokumen ini.

Ada berbagai macam taktik yang lebih khusus mengenai penggunaan kerangka kerja Scrum yang tidak dijelaskan di dalam dokumen ini.

Penggunaan Scrum

Scrum dikembangkan untuk mengelola dan mengembangkan produk. Scrum mulai digunakan sejak dan telah digunakan secara meluas di seluruh dunia, untuk:

1. Meneliti dan menggali potensi pasar, teknologi, dan kemampuan produk;
2. Mengembangkan produk dan peningkatan-peningkatannya;
3. Merilis produk dan peningkatan-peningkatannya, sesering mungkin di setiap hari;
4. Mengembangkan dan memelihara operasional sistem komputasi awan (daring, keamanan, sesuai permintaan) dan lingkungan operasional lain untuk penggunaan produk; dan,
5. Mengelola dan memperbaiki sebuah produk.

Scrum telah digunakan untuk mengembangkan perangkat lunak, perangkat keras, perangkat lunak terintegrasi, aplikasi dalam jaringan yang saling berinteraksi, kendaraan tanpa awak, sekolah, pemerintahan, pemasaran, mengelola operasional organisasi dan hampir semua hal yang kita gunakan di kehidupan sehari-hari sebagai seorang individu dan anggota masyarakat.

Keampuan Scrum dalam menghadapi kompleksitas semakin terbukti setiap harinya seiring dengan semakin meningkatnya kompleksitas dan interaksi antara teknologi, pasar dan lingkungan.

Scrum terbukti efektif dalam transfer pengetahuan secara berkala dan berkelanjutan. Scrum saat ini sudah digunakan secara luas untuk produk-produk, layanan-layanan, dan manajemen perusahaan induk.

Esensi dari Scrum adalah sebuah tim kecil yang terdiri dari beberapa orang. Tim ini bersifat sangat fleksibel dan mampu beradaptasi. Kekuatan ini terus berlanjut dalam satu tim, beberapa tim, banyak tim, maupun banyak tim yang berhubungan dalam mengembangkan, merilis, mengoperasikan dan menjaga pekerjaan; dan produk hasil pekerjaan dari ribuan orang. Mereka berkolaborasi dan saling berinteraksi melalui arsitektur pengembangan dan target lingkungan rilis produk yang mutakhir.

Pada saat kata “mengembangkan” dan “pengembangan” digunakan di dalam Panduan Scrum ini, keduanya mengacu pada pekerjaan yang kompleks seperti tipe pekerjaan yang telah dipaparkan di atas.

Teori Scrum

Scrum dibangun di atas teori proses kontrol empiris atau bisa disebut empirisme. Empirisme menyatakan bahwa pengetahuan datang dari pengalaman dan pengambilan keputusan didasari oleh apa yang telah diketahui hingga saat ini. Scrum menggunakan pendekatan yang bertahap dan berkelanjutan untuk mengoptimalkan kemampuan prediksi dan mengendalikan risiko. Tiga pilar yang memperkuat setiap implementasi dari proses kontrol empiris adalah: transparansi, inspeksi dan adaptasi.

Transparansi

Aspek signifikan dari sebuah proses harus dapat dilihat oleh orang-orang yang bertanggung jawab terhadap dampaknya. Transparansi membutuhkan aspek-aspek tersebut ditentukan oleh standar baku sehingga para pengamat memiliki pemahaman yang sama terhadap apa yang sedang ditinjau.

Sebagai contoh:

- Semua peserta harus memiliki pemahaman yang sama terkait istilah yang mengacu pada proses; dan,
- Mereka yang melakukan pekerjaan dan memeriksa *Increment* harus memiliki definisi "Selesai" yang sama.

Inspeksi

Pengguna Scrum harus sering menginspeksi artefak Scrum dan perkembangan menuju *Sprint Goal* agar mereka dapat mendeteksi adanya variansi hasil yang tidak diharapkan. Proses inspeksi juga disarankan tidak dilakukan terlalu sering sampai menghambat pekerjaan. Inspeksi akan sangat bermanfaat jika dilakukan oleh pemeriksa yang kompeten di saat dimana pekerjaan tersebut sedang berada.

Adaptasi

Jika pemeriksa menemukan bahwa ada satu hal atau lebih dari proses yang menyimpang di luar ambang batas yang bisa diterima yang dapat menyebabkan produk tidak bisa diterima, maka proses atau materi yang sedang diproses harus diubah. Perubahan harus dilakukan secepatnya untuk meminimalkan penyimpangan yang semakin jauh.

Scrum memiliki empat acara formal untuk melakukan inspeksi dan adaptasi, seperti yang dijabarkan di dalam dokumen ini yakni:

- *Sprint Planning*
- *Daily Scrum*
- *Sprint Review*
- *Sprint Retrospective*

Tata Nilai Scrum

Saat tata nilai dari komitmen, keberanian, fokus, keterbukaan, dan respek diwujudkan dan hidup di dalam *Scrum Team*, maka pilar-pilar Scrum seperti: Transparansi, Inspeksi dan Adaptasi akan menjadi hidup dan membangun rasa percaya satu sama lain. Anggota *Scrum Team* belajar dan menjiwai tata nilai tersebut bersama dengan penggunaan peran-peran, acara-acara dan artefak-artefak Scrum pada saat mereka bekerja.

Keberhasilan penggunaan Scrum bergantung pada orang-orang yang semakin menjiwai kelima tata nilai tersebut. Orang-orang secara pribadi berkomitmen untuk menggapai tujuan dari *Scrum Team*. Anggota *Scrum Team* memiliki keberanian untuk melakukan hal yang terbaik dan bekerja diatas masalah-masalah yang sulit. Semua pihak fokus pada pekerjaan di dalam *Sprint* dan tujuan-tujuan dari *Scrum Team*. *Scrum Team* dan pihak-pihak yang berkepentingan setuju untuk terbuka terhadap semua pekerjaan dan tantangan di dalam melakukan pekerjaan. Anggota *Scrum Team* saling respek satu sama lain bahwasanya mereka adalah orang-orang yang memiliki kemampuan dan kemandirian.

Scrum Team

Scrum Team terdiri dari *Product Owner*, *Development Team* dan *Scrum Master*. *Scrum Team* bersifat swakelola dan lintas-fungsi. Tim yang swakelola memilih cara terbaik dalam mengerjakan pekerjaan mereka, bukan diperintah oleh orang lain di luar tim ini. Tim yang lintas-fungsi memiliki semua keahlian yang dibutuhkan untuk menyelesaikan pekerjaan mereka tanpa bergantung pada orang lain di luar tim ini. Bentuk tim dalam Scrum dirancang untuk mengoptimalkan fleksibilitas, kreativitas dan produktivitas. Bentuk *Scrum Team* telah terbukti menjadikan tim semakin efektif dalam mengerjakan semua tipe pekerjaan yang telah disebutkan sebelumnya dan untuk jenis pekerjaan kompleks apa pun.

Scrum Team menghantarkan produk secara iteratif dan inkremental guna memaksimalkan peluang untuk mendapatkan umpan balik. Penghantaran produk “Selesai” secara inkremental dilakukan guna memastikan versi produk yang berpotensi untuk digunakan selalu siap tersedia.

Product Owner

Product Owner adalah orang yang bertanggung jawab untuk memaksimalkan nilai bisnis dari produk yang dihasilkan oleh *Development Team*. Cara melakukannya sangat bervariasi antar organisasi, *Scrum Team* dan individu.

Product Owner adalah satu-satunya orang yang bertanggung jawab dalam pengelolaan *Product Backlog*. Pengelolaan *Product Backlog* termasuk:

- Menyampaikan isi dari *Product Backlog item* secara jelas;
- Mengurutkan *Product Backlog item* untuk mencapai tujuan dan misi dengan cara terbaik;
- Mengoptimalkan nilai bisnis dari pekerjaan yang dilakukan oleh *Development Team*;
- Memastikan agar *Product Backlog* dapat dilihat, transparan, dan jelas untuk semua pihak, dan menampilkan apa yang akan dikerjakan selanjutnya oleh *Scrum Team*; dan,
- Memastikan *Development Team* memahami *Product Backlog item* hingga batas tertentu.

Product Owner dapat melakukan semua pekerjaan di atas atau meminta *Development Team* untuk mengerjakannya. Namun, hanya *Product Owner* yang bertanggung gugat terhadap *Product Backlog*.

Product Owner hanya satu orang saja dan bukan berupa komite. *Product Owner* dapat mewakili keinginan dari komite di dalam *Product Backlog*, tetapi pihak-pihak yang ingin mengubah prioritas dari *Product Backlog item* harus menyampaikannya lewat *Product Owner*.

Agar *Product Owner* dapat berhasil, seluruh organisasi harus menghargai keputusannya. Keputusannya dapat dilihat dari isi dan urutan *Product Backlog*. Tidak ada siapapun yang dapat memaksa *Development Team* untuk bekerja selain dari yang tercantum di dalam *Product Backlog*.

Development Team

Development Team terdiri dari para ahli profesi yang bekerja untuk menghantarkan *Increment* “Selesai” yang berpotensi untuk dirilis di setiap akhir *Sprint*. *Increment* “Selesai” wajib tersedia pada saat *Sprint Review*. Hanya anggota dari *Development Team* yang membuat *Increment* ini.

Development Team dibentuk dan diberikan wewenang oleh organisasi untuk menyusun dan mengelola pekerjaan mereka sendiri. Hasil sinergi dari tim ini akan mengoptimalkan efisiensi dan efektivitas *Development Team* secara keseluruhan.

Development Team memiliki karakteristik sebagai berikut:

- Mereka swakelola. Tidak ada satu orang pun (bahkan *Scrum Master*) yang memberitahu *Development Team* bagaimana memanifestasikan *Product Backlog* menjadi gabungan fungsionalitas yang berpotensi untuk dirilis;
- *Development Team* bersifat lintas-fungsi, mereka memiliki semua keahlian yang diperlukan untuk membuat *Increment*;
- *Scrum* tidak mengenal jabatan untuk anggota *Development Team*, terlepas dari jenis pekerjaan yang mereka lakukan;
- Terlepas dari jenis pekerjaan yang perlu dilakukan, misal testing, arsitektur, operasional, ataupun analisa bisnis, *Scrum* tidak mengenal pengelompokan di dalam *Development Team* berdasarkan jenis-jenis pekerjaan ini; dan,
- Setiap anggota *Development Team* bisa saja memiliki keahlian khusus dan fokus di bidang tertentu, tetapi tanggung gugat adalah milik seluruh anggota *Development Team*.

Jumlah Anggota *Development Team*

Jumlah anggota *Development Team* yang optimal adalah tidak terlalu banyak agar tim ini tetap lincah, tetapi cukup banyak untuk dapat menyelesaikan pekerjaan secara signifikan di dalam satu *Sprint*. Kurang dari tiga orang akan mengurangi interaksi dan menyebabkan produktivitas yang rendah dan dapat menghadapkan tim pada kekurangan keahlian yang dibutuhkan di dalam *Sprint*, sehingga *Development Team* tidak dapat menghantarkan *Increment* yang berpotensi untuk dirilis. Lebih dari sembilan orang menyebabkan terlalu banyaknya koordinasi dan terlalu banyak kompleksitas sehingga proses empiris bisa menjadi tidak bermanfaat. *Product Owner* dan *Scrum Master* tidak termasuk dalam jumlah ini kecuali mereka juga turut mengerjakan pekerjaan dari *Sprint Backlog*.

Scrum Master

Scrum Master bertanggung jawab untuk mengenalkan dan menyokong penggunaan Scrum sebagaimana dijelaskan di dalam Panduan Scrum ini. *Scrum Master* melakukan ini dengan membantu orang-orang agar dapat memahami teori, praktik-praktik, aturan-aturan dan tata nilai Scrum.

Scrum Master adalah pemimpin yang melayani *Scrum Team*. *Scrum Master* membantu orang-orang di luar *Scrum Team* untuk dapat memahami interaksi mana yang bermanfaat dan tidak bermanfaat. *Scrum Master* membantu orang-orang untuk mengubah interaksi ini guna memaksimalkan nilai bisnis yang dihasilkan oleh *Scrum Team*.

Layanan *Scrum Master* kepada *Product Owner*

Scrum Master melayani *Product Owner* lewat beberapa cara, termasuk:

- Memastikan tujuan, ruang lingkup dan ranah produk dipahami sebaik mungkin oleh semua orang di dalam *Scrum Team*;
- Menemukan teknik yang paling efektif untuk mengelola *Product Backlog*;
- Membantu *Scrum Team* untuk memahami perlunya *Product Backlog item* yang jelas dan padat;
- Memahami perencanaan produk di dalam lingkungan empiris;
- Memastikan *Product Owner* paham bagaimana mengelola *Product Backlog* yang memaksimalkan nilai bisnis;
- Memahami dan mempraktikkan *agility*; dan,
- Memfasilitasi acara-acara Scrum bila diminta atau dibutuhkan;

Layanan *Scrum Master* kepada *Development Team*

Scrum Master melayani *Development Team* lewat beberapa cara, termasuk:

- Membimbing *Development Team* agar dapat swakelola dan lintas-fungsi;
- Membantu *Development Team* untuk menghasilkan produk bernilai bisnis tinggi;
- Menghilangkan hambatan yang memperlambat perkembangan pekerjaan *Development Team*;
- Memfasilitasi acara-acara Scrum bila diminta atau dibutuhkan; dan,
- Membimbing *Development Team* di organisasi dimana Scrum belum sepenuhnya dipraktikkan dan dipahami.

Layanan *Scrum Master* kepada Organisasi

Scrum Master melayani organisasi lewat beberapa cara, termasuk:

- Memimpin dan membimbing organisasi dalam penggunaan *Scrum*;
- Merencanakan implementasi *Scrum* di dalam organisasi;
- Membantu pegawai dan pemilik kepentingan untuk memahami dan menggunakan *Scrum* dan pengembangan produk secara empiris;
- Membuat perubahan yang dapat meningkatkan produktivitas *Scrum Team*; dan,
- Bekerja dengan *Scrum Master* lainnya untuk meningkatkan efektivitas dari penggunaan *Scrum* di dalam organisasi.

Acara-acara *Scrum*

Acara-acara wajib dalam *Scrum* diselenggarakan guna terciptanya kerutinan dan mengurangi pertemuan lain yang bukan merupakan bagian dari *Scrum*. Setiap acara memiliki batasan waktu, artinya setiap acara memiliki durasi waktu maksimal. Ketika *Sprint* dimulai, durasinya tetap yang artinya tidak bisa diperpendek maupun diperpanjang. Acara selain *Sprint* dapat berakhir kapanpun juga ketika tujuan dari acara tersebut telah tercapai, hal ini dilakukan guna memastikan waktu yang digunakan tidak berlebihan dan tidak ada waktu yang terbuang di sepanjang proses.

Selain *Sprint* itu sendiri, yang merupakan wadah untuk acara-acara lainnya, setiap acara di dalam *Scrum* adalah sebuah kesempatan formal untuk menginspeksi dan mengadaptasi sesuatu. Acara-acara ini secara khusus dirancang untuk mewujudkan transparansi dan inspeksi yang bersifat kritis. Tidak diselenggarakannya acara-acara ini akan mengurangi transparansi dan menghilangkan kesempatan untuk melakukan inspeksi dan adaptasi.

Sprint

Jantung dari *Scrum* adalah *Sprint*, yaitu sebuah batasan waktu dengan durasi satu bulan atau kurang, dimana terdapat proses pembuatan *Increment* yang “Selesai”, dapat digunakan dan berpotensi untuk dirilis. *Sprint* memiliki durasi yang konsisten sepanjang daur hidup pengembangan produk. *Sprint* yang baru langsung dimulai setelah *Sprint* sebelumnya selesai.

Sprint berisi dan terdiri dari *Sprint Planning*, *Daily Scrum*, pengembangan produk, *Sprint Review* dan *Sprint Retrospective*.

Pada saat *Sprint* berjalan:

- Tidak boleh ada perubahan yang dapat mengancam *Sprint Goal*;
- Tingkat kualitas tidak boleh menurun; dan,
- Ruang lingkup dapat diklarifikasi dan dinegosiasi ulang antara *Product Owner* dan *Development Team* setiap kali adanya hal baru yang mereka pelajari;

Setiap *Sprint* bisa dianggap sebagai sebuah proyek dengan durasi tidak lebih dari satu bulan yang dijalankan untuk mencapai sebuah tujuan. Setiap *Sprint* memiliki tujuan mengenai apa yang harus dibangun, sebuah rancangan dan perencanaan fleksibel yang memandu pembangunan tersebut, daftar pekerjaan, dan hasil dari *Increment*.

Sprint dibatasi tidak lebih dari satu bulan kalender. Ketika durasi *Sprint* terlalu lama maka definisi dari apa yang akan dikembangkan dapat berubah, kompleksitas dapat meningkat, dan risiko juga dapat meningkat. Dengan memastikan adanya inspeksi dan adaptasi, keberadaan *Sprint* menciptakan sebuah tingkat kemungkinan perkembangan pekerjaan menuju *Sprint Goal* setidaknya setiap satu bulan kalender. Dengan adanya *Sprint*, risiko pengeluaran biaya dibatasi menjadi maksimal satu bulan kalender.

Membatalkan *Sprint*

Sprint dapat dibatalkan sebelum mencapai batasan waktu. Hanya *Product Owner* yang memiliki otoritas untuk membatalkan *Sprint*, walaupun ia bisa saja membatalkannya karena dipengaruhi oleh pemegang kepentingan, *Development Team* ataupun *Scrum Master*.

Sprint bisa saja dibatalkan bila *Sprint Goal* berubah. Hal ini dapat terjadi ketika perusahaan mengubah haluan ataupun ketika pasar atau teknologi telah berubah. *Sprint* seharusnya dibatalkan bila sudah tidak masuk akal untuk dilanjutkan yang disebabkan oleh keadaan-keadaan terkini. Akan tetapi karena durasi *Sprint* yang singkat, pembatalan *Sprint* seringkali tidak logis untuk dilakukan.

Ketika *Sprint* dibatalkan, semua *Product Backlog item* yang tuntas dan “Selesai” ditinjau kembali. Bila sebagian dari hasil pekerjaan tersebut berpotensi untuk dirilis, maka biasanya *Product Owner* menerimanya. Semua pekerjaan yang belum tuntas diestimasi ulang dan dimasukkan kembali ke *Product Backlog*. Semua nilai bisnis di dalam pekerjaan yang selesai itu menurun secara cepat dan harus diestimasi ulang secara rutin.

Pembatalan *Sprint* memakan berbagai sumber daya karena semua orang harus dikelompokkan lagi di *Sprint Planning* untuk memulai *Sprint* yang lain. Pembatalan *Sprint* seringkali menimbulkan trauma di dalam *Scrum Team* dan sangat tidak umum dilakukan.

Sprint Planning

Pekerjaan yang akan dikerjakan di *Sprint* direncanakan pada saat *Sprint Planning*. Perencanaan ini dilakukan secara kolaboratif oleh seluruh anggota *Scrum Team*.

Sprint Planning memiliki batasan waktu maksimal delapan jam untuk *Sprint* yang berdurasi satu bulan. Untuk *Sprint* yang lebih singkat, acara ini biasanya lebih singkat. *Scrum Master* memastikan acara ini diselenggarakan dan peserta memahami tujuannya. *Scrum Master* mengedukasi *Scrum Team* untuk menjaganya di dalam batasan waktu.

Sprint Planning menjawab pertanyaan-pertanyaan berikut:

- Apa yang dapat dihantarkan ke dalam *Increment* dari *Sprint* ini?
- Bagaimana mereka bisa menyelesaikan pekerjaan yang dibutuhkan untuk menghantarkan *Increment*?

Topik Satu: Apa yang bisa diselesaikan di *Sprint* ini?

Development Team memprakirakan fungsionalitas yang bisa dikembangkan selama *Sprint*. *Product Owner* membahas obyektif yang harus dicapai di *Sprint* dan *Product Backlog item* yang dapat mencapai *Sprint Goal* bila diselesaikan. *Scrum Team* berkolaborasi untuk memahami seluruh pekerjaan untuk *Sprint*.

Masukan untuk pertemuan ini adalah *Product Backlog*, *Increment* terkini, proyeksi kapasitas *Development Team* di *Sprint* ini, dan kinerja sebelumnya dari *Development Team*. Jumlah pekerjaan yang dipilih dari *Product Backlog* untuk *Sprint* merupakan keputusan *Development Team* sepenuhnya. Hanya *Development Team* yang dapat menilai apa yang dapat mereka selesaikan di *Sprint*.

Pada saat *Sprint Planning*, *Scrum Team* juga menentukan *Sprint Goal*. *Sprint Goal* adalah objektif yang akan dicapai selama *Sprint* melalui implementasi *Product Backlog* dan menyediakan panduan untuk *Development Team* mengembangkan *Increment*.

Topik Dua: Bagaimana pekerjaan terpilih diselesaikan?

Dengan adanya *Sprint Goal* dan *Product Backlog item* terpilih untuk *Sprint*, *Development Team* memutuskan bagaimana mereka akan mengembangkan fungsionalitas ini menjadi *Increment* “Selesai” pada saat *Sprint*. *Product Backlog item* terpilih untuk *Sprint* ini beserta perencanaan untuk menghantarkan dinamakan *Sprint Backlog*.

Development Team biasanya memulai dengan merancang sistem dan pekerjaan yang perlu dikerjakan untuk mengubah *Product Backlog* menjadi *Increment* yang berfungsi. Ukuran pekerjaan atau estimasi untuk masing-masing pekerjaan bisa bervariasi. Pada saat *Sprint Planning* jumlah pekerjaan cukup tersedia agar *Development Team* dapat membuat prakiraan mengenai apa yang mereka yakini bisa diselesaikan di *Sprint*. Pekerjaan yang akan dikerjakan di beberapa hari pertama di *Sprint* dipecah hingga satuan satu hari atau kurang di sebelum pertemuan ini berakhir. *Development Team* swakelola untuk mengambil pekerjaan dari *Sprint Backlog* pada saat *Sprint Planning* maupun di sepanjang *Sprint*.

Product Owner dapat membantu memberi klarifikasi terhadap *Product Backlog item* terpilih dan menukar atau mengeluarkannya. Bila *Development Team* menganggap mereka memiliki terlalu banyak atau terlalu sedikit pekerjaan, mereka dapat menegosiasikan ulang *Product Backlog item* terpilih dengan *Product Owner*. *Development Team* juga dapat mengundang pihak lain untuk menyediakan saran perihal teknis atau bidang khusus.

Sebelum *Sprint Planning* berakhir, *Development Team* harus bisa menjelaskan kepada *Product Owner* dan *Scrum Master* bagaimana mereka akan bekerja sebagai tim swakelola untuk mencapai *Sprint Goal* dan membuat *Increment* yang diharapkan.

Sprint Goal

Sprint Goal adalah sebuah objektif untuk *Sprint* yang dapat dicapai lewat pengimplementasian *Product Backlog*. *Sprint Goal* merupakan panduan bagi *Development Team* untuk menjawab pertanyaan mengapa mereka mengembangkan *Increment*. *Sprint Goal* dibuat pada pertemuan *Sprint Planning*. *Sprint* memberikan ruang fleksibilitas mengenai fungsionalitas yang akan diimplementasikan di dalam *Sprint*. *Product Backlog item* terpilih merupakan satu fungsi yang selaras yang bisa jadi *Sprint Goal*. *Sprint Goal* bisa saja menghubungkan pekerjaan yang tidak memiliki keterkaitan agar *Development Team* tidak bekerja dari instruksi kerja yang berbeda-beda.

Development Team selalu mengingat *Sprint Goal* yang telah ditetapkan pada saat mereka bekerja. *Development Team* mengimplementasikan fungsionalitas dan teknologi guna mencapai *Sprint Goal* tersebut. Apabila pekerjaannya ternyata berbeda dengan apa yang diharapkan oleh *Development Team* maka mereka akan berkolaborasi dengan *Product Owner* untuk menegosiasikan ruang lingkup *Sprint Backlog* di dalam *Sprint*.

Daily Scrum

Daily Scrum adalah acara untuk *Development Team* yang memiliki batasan waktu 15 menit. Acara ini dilakukan setiap hari selama *Sprint* berlangsung. Di acara ini, *Development Team* membuat rencana kerja untuk 24 jam ke depan. Acara ini mengoptimalkan kolaborasi dan performa dari tim dengan melakukan inspeksi pada pekerjaan yang dilakukan semenjak *Daily Scrum* sebelumnya dan melakukan prakiraan terhadap pekerjaan selanjutnya di dalam *Sprint*. *Daily Scrum* dilakukan di waktu dan tempat yang sama setiap harinya untuk mengurangi kompleksitas.

Development Team menggunakan *Daily Scrum* untuk menginspeksi perkembangan pekerjaan menuju *Sprint Goal* dan tren perkembangan penyelesaian pekerjaan di *Sprint Backlog*. *Daily Scrum* meningkatkan kemungkinan *Development Team* untuk mencapai *Sprint Goal*. Setiap hari, *Development Team* harus memahami bagaimana mereka bekerjasama sebagai tim swakelola untuk mencapai *Sprint Goal* dan membuat *Increment* yang diharapkan di akhir *Sprint*.

Struktur dari pertemuan ini ditentukan oleh *Development Team* dan bisa diadakan lewat berbagai macam cara selama pertemuan ini fokus terhadap kemajuan menuju *Sprint Goal*. Beberapa *Development Team* akan menggunakan pertanyaan-pertanyaan, beberapa akan lebih berdiskusi. Berikut adalah contoh pertanyaan yang mungkin saja digunakan:

- Apa yang telah saya lakukan kemarin untuk membantu *Development Team* mencapai *Sprint Goal*?
- Apa yang akan saya lakukan hari ini untuk membantu *Development Team* mencapai *Sprint Goal*?
- Apakah saya melihat ada hambatan yang menghalangi saya ataupun *Development Team* dalam mencapai *Sprint Goal*?

Development Team atau beberapa anggota tim seringkali berkumpul setelah *Daily Scrum* untuk diskusi yang lebih mendalam atau melakukan adaptasi atau melakukan rencana ulang terhadap sisa pekerjaan dari *Sprint*.

Scrum Master memastikan bahwa *Development Team* menyelenggarakan pertemuan tersebut, tetapi *Development Team* yang bertanggung jawab untuk menjalankan *Daily Scrum*. *Scrum Master* mengajari *Development Team* untuk menjaga *Daily Scrum* tetap di dalam batasan waktu 15 menit.

Daily Scrum adalah pertemuan internal untuk *Development Team*. Jika orang lain hadir, *Scrum Master* memastikan mereka tidak mengganggu jalannya pertemuan ini.

Daily Scrum meningkatkan kualitas komunikasi, mengeliminasi pertemuan-pertemuan lain, mengidentifikasi hambatan untuk dihilangkan, menyoroti dan mendukung pengambilan keputusan secara cepat, meningkatkan tingkat pengetahuan dari *Development Team*. Hal ini merupakan kunci dari pertemuan inspeksi dan adaptasi.

Sprint Review

Sprint Review diselenggarakan di akhir *Sprint* untuk menginspeksi *Increment* dan mengadaptasi *Product Backlog* bila diperlukan. Pada saat *Sprint Review*, *Scrum Team* dan pemegang kepentingan berkolaborasi untuk meninjau apa yang sudah diselesaikan di *Sprint*. Berdasarkan hasil tinjauan tersebut dan perubahan terhadap *Product Backlog* di dalam *Sprint*, hadirin berkolaborasi untuk menentukan pekerjaan selanjutnya yang dapat dilakukan untuk mengoptimalkan nilai bisnis. Ini adalah pertemuan informal, bukan pertemuan laporan status, dan presentasi *Increment* dilakukan guna mendapatkan umpan balik dan mengembangkan kemampuan kolaborasi.

Pertemuan ini paling lama diselenggarakan selama empat jam untuk *Sprint* berdurasi satu bulan. Untuk *Sprint* yang lebih singkat, durasi acara ini biasanya lebih singkat. *Scrum Master* memastikan acara ini terselenggara dan hadirin memahami tujuannya. *Scrum Master* mengedukasi setiap hadirin untuk menjaganya di dalam batasan waktu.

Sprint Review mengandung unsur-unsur berikut:

- Hadirin yang di dalamnya adalah *Scrum Team* dan para pemegang kepentingan utama yang diundang oleh *Product Owner*;
- *Product Owner* menjelaskan *Product Backlog item* yang sudah “Selesai” dan yang belum “Selesai”;
- *Development Team* menjelaskan apa yang berjalan dengan baik sepanjang *Sprint*, masalah-masalah yang mereka hadapi dan bagaimana cara memecahkannya;
- *Development Team* mendemonstrasikan pekerjaan yang telah mereka “Selesai”-kan dan menjawab pertanyaan-pertanyaan mengenai *Increment*;
- *Product Owner* menjelaskan keadaan *Product Backlog* hingga saat ini. Ia juga memproyeksikan target dan tanggal penghantaran produk berdasarkan perkembangan hingga hari ini (bila ditanyakan);
- Seluruh hadirin berkolaborasi mengenai apa yang akan mereka lakukan selanjutnya, artinya *Sprint Review* menghasilkan masukan berharga untuk *Sprint Planning* berikutnya;
- Meninjau bagaimana keadaan pasar atau potensi penggunaan produk, yang mungkin telah berubah dan hal apa yang paling bernilai untuk dikerjakan berikutnya; dan,
- Meninjau jangka waktu, anggaran, potensi kapabilitas, dan keadaan pasar untuk rilis fitur atau kemampuan produk yang sudah ditargetkan.

Hasil dari *Sprint Review* adalah *Product Backlog* yang sudah direvisi yang menjabarkan *Product Backlog item* yang mungkin diimplementasikan di *Sprint* berikutnya. *Product Backlog* juga dapat disesuaikan secara keseluruhan untuk mendapatkan peluang baru di pasar.

Sprint Retrospective

Sprint Retrospective adalah sebuah kesempatan bagi *Scrum Team* untuk menginspeksi dirinya sendiri dan membuat perencanaan mengenai peningkatan yang akan dilakukan di *Sprint* berikutnya.

Sprint Retrospective terselenggara setelah *Sprint Review* dan sebelum *Sprint Planning* berikutnya. Acara ini diselenggarakan paling lama tiga jam untuk *Sprint* yang berdurasi satu bulan. Untuk *Sprint* yang lebih singkat, durasi acara ini biasanya lebih singkat. *Scrum Master* memastikan acara ini terselenggara dan setiap peserta memahami tujuannya.

Scrum Master memastikan bahwa acara ini bernuansa positif dan produktif. *Scrum Master* mengedukasi peserta untuk menjaganya di dalam batasan waktu. *Scrum Master* turut berpartisipasi sebagai rekan kerja sejawat yang bertanggung gugat terhadap proses Scrum di pertemuan ini.

Tujuan dari *Sprint Retrospective* adalah:

- Menginspeksi bagaimana jalannya *Sprint* terakhir yang terkait dengan orang-orang, hubungan antar mereka, proses, dan alat-alat yang digunakan;
- Mengidentifikasi dan mengurutkan hal utama yang berjalan dengan baik dan peningkatan yang berpotensi untuk dilakukan; dan,
- Membuat perencanaan untuk implementasi peningkatan cara kerja *Scrum Team*.

Scrum Master mendorong *Scrum Team* untuk membuat peningkatan dalam proses kerangka kerja Scrum, proses dan praktik-praktik pengembangan produk agar *Sprint* berikutnya lebih efektif dan menyenangkan. Di setiap *Sprint Retrospective*, *Scrum Team* merencanakan cara-cara untuk meningkatkan kualitas produk dengan cara meningkatkan proses kerja ataupun mengubah isi dari definisi “Selesai”, apabila hal tersebut tidak bersinggungan dengan standar pengembangan produk di organisasi.

Sebelum *Sprint Retrospective* berakhir, *Scrum Team* harus menyepakati peningkatan yang akan mereka implementasikan di *Sprint* berikutnya. Mengimplementasikan peningkatan di *Sprint* berikutnya adalah bentuk inspeksi dan adaptasi terhadap *Scrum Team* itu sendiri. Meskipun peningkatan dapat dilakukan kapanpun di sepanjang *Sprint*, *Sprint Retrospective* adalah sebuah kesempatan formal yang fokus pada inspeksi dan adaptasi.

Artefak-artefak Scrum

Artefak Scrum merepresentasikan pekerjaan atau nilai bisnis guna terciptanya transparansi dan kesempatan untuk menginspeksi dan mengadaptasi. Artefak-artefak yang dijabarkan Scrum dirancang sedemikian rupa untuk memaksimalkan transparansi informasi utama agar setiap orang memiliki pemahaman yang sama mengenai artefak tersebut.

Product Backlog

Product Backlog adalah daftar terurut semua hal yang telah diketahui hingga saat ini harus ada di dalam produk. *Product Backlog* adalah satu-satunya sumber kebutuhan untuk semua perubahan yang perlu diberlakukan terhadap produk. *Product Owner* bertanggung jawab terhadap *Product Backlog*, termasuk isi, ketersediaan dan urutannya.

Sebuah *Product Backlog* tidak pernah tuntas. Di awal pengembangan produk, *Product Backlog* berisi daftar kebutuhan yang diketahui dan dipahami saat ini. *Product Backlog* berevolusi seiring dengan perkembangan produk dan lingkungan dimana produk tersebut digunakan. *Product Backlog* bersifat dinamis dan berubah terus secara konstan agar produk menjadi layak, kompetitif dan bermanfaat. Selama produk masih ada, *Product Backlog* juga akan selalu ada.

Product Backlog adalah daftar dari seluruh fitur, fungsi, kebutuhan, peningkatan, dan perbaikan yang perlu diberlakukan terhadap produk pada rilis mendatang. *Product Backlog item* memiliki atribut deskripsi, urutan, estimasi dan nilai bisnis. *Product Backlog item* seringkali berisi deskripsi pengujian produk yang akan menjadi bukti ketuntasan ketika *Product Backlog item* tersebut dikatakan “Selesai”.

Seiring produk digunakan serta menghasilkan nilai bisnis dan didapatkannya umpan balik dari pasar, *Product Backlog* semakin berkembang dan isinya bertambah banyak. Daftar kebutuhan tidak pernah berhenti berubah, jadi dapat dikatakan kalau *Product Backlog* adalah artefak hidup. Perubahan dalam kebutuhan bisnis, kondisi pasar ataupun teknologi dapat mengubah *Product Backlog*.

Beberapa *Scrum Team* seringkali mengembangkan satu produk yang sama. Satu *Product Backlog* digunakan untuk mendeskripsikan pekerjaan yang akan datang terhadap satu produk ini. Atribut *Product Backlog* yang mengelompokkan beberapa *Product Backlog item* dapat diberlakukan.

Penyempurnaan *Product Backlog* adalah kegiatan menambahkan detail, estimasi, dan urutan terhadap *Product Backlog item*. Kegiatan ini adalah proses berkesinambungan dimana *Product Owner* dan *Development Team* berkolaborasi untuk mendetilkkan *Product Backlog item*. Pada kegiatan penyempurnaan *Product Backlog*, *Product Backlog item* ditinjau dan direvisi. *Scrum Team* yang memutuskan kapan dan bagaimana kegiatan penyempurnaan ini dilakukan. Penyempurnaan biasanya memakan waktu tidak lebih dari 10% kapasitas *Development Team*. Namun, *Product Backlog item* dapat diperbarui kapanpun juga oleh *Product Owner* ataupun berdasarkan kebijakan *Product Owner*.

Urutan teratas dalam *Product Backlog item* biasanya lebih jelas dan lebih detail dibandingkan dengan *Product Backlog item* di urutan lebih bawah. Estimasi yang semakin akurat dilandasi oleh tingkat kejelasan dan kedetilan yang semakin tinggi; semakin bawah urutannya semakin rendah tingkat kedetilannya. *Product Backlog item* yang akan menyibukkan *Development Team* selama *Sprint* disempurnakan supaya setidaknya ada satu *Product Backlog item* yang bisa “Selesai” dalam *Sprint*. *Product Backlog item* yang bisa di-”Selesai”-kan oleh *Development Team* dalam satu *Sprint* “Siap” untuk dipilih pada saat *Sprint Planning*. Hasil dari aktivitas penyempurnaan ini biasanya menyebabkan transparansi *Product Backlog item* menjadi lebih tinggi.

Development Team bertanggung jawab untuk seluruh estimasi. *Product Owner* dapat mempengaruhi *Development Team* dengan cara membantu mereka untuk memahami dan membuat pertukaran, tetapi pada akhirnya orang-orang yang melakukan pekerjaan yang menentukan estimasi akhir.

Memantau Perkembangan Menuju Tujuan Akhir

Total sisa pekerjaan menuju tujuan akhir dapat dijumlahkan kapanpun juga. *Product Owner* memantau total sisa pekerjaan ini setidaknya di setiap *Sprint Review*. *Product Owner* membandingkannya dengan jumlah sisa pekerjaan dari *Sprint Review* sebelumnya untuk mengkaji proyeksi perkembangan pekerjaan menuju tujuan akhir di waktu yang telah ditetapkan. Informasi ini dibuat transparan untuk semua pemilik kepentingan.

Berbagai praktik proyeksi tren untuk memprakirakan perkembangan telah digunakan oleh praktisi Scrum seperti misalnya *burn-downs*, *burn-ups* ataupun *cumulative flows*. Praktik-praktik ini telah terbukti berguna. Namun, praktik-praktik ini tidak menggantikan pentingnya empirisme. Dalam lingkungan kompleks, apa yang akan terjadi di masa mendatang tidak bisa diketahui sebelumnya. Hanya apa yang telah terjadi di masa lalu yang dapat digunakan sebagai pembuatan keputusan jauh ke depan.

Sprint Backlog

Sprint Backlog adalah daftar *Product Backlog item* yang terpilih untuk *Sprint* ditambah perencanaan untuk menghantarkan *Increment* dan mencapai *Sprint Goal*. *Sprint Backlog* adalah prakiraan dari *Development Team* mengenai fungsionalitas yang akan masuk ke dalam *Increment* berikutnya dan pekerjaan yang perlu dikerjakan untuk menghantarkan fungsionalitasnya menjadi *Increment* yang “Selesai”.

Sprint Backlog menampilkan seluruh pekerjaan yang menurut *Development Team* perlu dikerjakan untuk mencapai *Sprint Goal*. Untuk memastikan adanya peningkatan berkelanjutan, *Sprint Backlog* berisi setidaknya satu peningkatan proses dengan prioritas tertinggi dari hasil pertemuan *Retrospective Sprint* sebelumnya.

Sprint Backlog adalah perencanaan yang cukup rinci sehingga perubahan yang sedang dikerjakan dapat dipahami pada saat *Daily Scrum*. *Development Team* mengubah *Sprint Backlog* sepanjang *Sprint*, dan *Sprint Backlog* muncul saat *Sprint* berlangsung. Seiring dengan *Development Team* bekerja sesuai perencanaan dan semakin bertambahnya pemahaman mereka mengenai pekerjaan yang dibutuhkan untuk mencapai *Sprint Goal*, *Sprint Backlog* baru bermunculan.

Jika ada pekerjaan baru yang diperlukan, maka *Development Team* menambahkannya ke *Sprint Backlog*. Jika ada yang sudah diselesaikan maka estimasi sisa pekerjaannya diperbarui. Jika ternyata ada pekerjaan yang tidak diperlukan, maka dihilangkan. Hanya *Development Team* yang dapat mengubah isi dari *Sprint Backlog* milik mereka sepanjang *Sprint*. *Sprint Backlog* dapat dilihat secara jelas, menggambarkan keadaan terkini mengenai sisa pekerjaan yang telah direncanakan *Development Team* untuk diselesaikan dalam *Sprint* dan merupakan hak milik *Development Team* sepenuhnya.

Memantau Perkembangan *Sprint*

Di setiap saat di dalam *Sprint*, total sisa pekerjaan di dalam *Sprint Backlog* dapat dijumlahkan. *Development Team* meninjau total sisa pekerjaan ini setidaknya setiap hari pada saat *Daily Scrum* untuk dapat memproyeksikan apakah kira-kira *Sprint Goal* dapat tercapai. Dengan memantau sisa pekerjaan ini sepanjang *Sprint*, *Development Team* dapat mengelola perkembangan pekerjaan mereka.

Increment

Increment adalah manifestasi dari *Product Backlog item* yang diselesaikan dalam *Sprint* dan total nilai bisnis *Increment* dari seluruh *Sprint* yang lalu. Di akhir *Sprint*, *Increment* yang baru harus “Selesai”, yang artinya *Increment* tersebut harus berada pada kondisi yang dapat digunakan dan sesuai dengan definisi “Selesai” milik *Scrum Team*. Sebuah *Increment* adalah hasil pekerjaan yang bisa diinspeksi dan telah selesai guna mendukung empirisme di akhir *Sprint*. *Increment* adalah sebuah langkah kecil menuju sebuah visi ataupun tujuan. *Increment* harus bersifat dapat digunakan terlepas apakah *Product Owner* memutuskan untuk merilisnya atau tidak.

Transparansi Artefak

Scrum mengandalkan transparansi. Keputusan untuk mengoptimalkan nilai bisnis dan mengendalikan risiko dibuat berdasarkan keadaan terkini dari artefak-artefak Scrum. Keputusan-keputusan ini akan mempunyai dasar yang kuat apabila transparansi sudah utuh. Apabila artefak-artefak Scrum tidak terlalu transparan maka keputusan kurang dapat dipercaya, nilai bisnis bisa jadi menurun dan risiko pun bisa jadi meningkat.

Scrum Master harus bekerja-sama dengan *Product Owner*, *Development Team* dan pihak-pihak lainnya untuk memastikan bahwa artefak-artefak Scrum benar-benar transparan. Ada beberapa praktik untuk mengatasi transparansi yang tidak lengkap dan seorang *Scrum Master* harus membantu orang-orang untuk menerapkan praktik-praktik yang paling sesuai bila tidak terjadi transparansi. Seorang *Scrum Master* dapat mendeteksi transparansi yang tidak utuh dengan menginspeksi artefak-artefak, merasakan terjadinya pola yang berulang-ulang, mendengarkan dengan seksama apa yang dikatakan orang-orang dan mendeteksi perbedaan antara apa yang diharapkan dengan apa yang dihasilkan.

Tugas seorang *Scrum Master* adalah bekerja-sama dengan *Scrum Team* dan organisasi untuk meningkatkan transparansi dari artefak-artefak Scrum. Pekerjaan ini melibatkan belajar, meyakinkan orang-orang dan membuat perubahan. Transparansi tidak terjadi dalam semalam, tetapi merupakan sebuah perjalanan jangka panjang.

Definisi “Selesai”

Ketika *Product Backlog item* atau sebuah *Increment* dikatakan “Selesai”, semua orang harus memahami apa artinya “Selesai”. Walaupun definisi ini bisa berbeda jauh antar *Scrum Team*, setiap anggotanya harus memiliki pemahaman yang sama mengenai apa artinya ketika pekerjaan dikatakan tuntas untuk memastikan adanya transparansi. Hal ini dinamakan definisi “Selesai” untuk *Scrum Team* dan digunakan untuk menilai kapan pekerjaan terhadap *Increment* dapat dikatakan tuntas.

Definisi yang sama membimbing *Development Team* untuk mengetahui berapa banyak jumlah *Product Backlog item* yang dapat mereka pilih pada saat *Sprint Planning*. Tujuan dari setiap *Sprint* adalah untuk menghantarkan fungsionalitas yang berpotensi untuk dirilis, sesuai dengan definisi “Selesai” milik *Scrum Team* saat ini.

Development Team menghantarkan *Increment* yang berfungsi setiap *Sprint*. *Increment* ini dapat digunakan, yang artinya *Product Owner* dapat memutuskan untuk segera merilisnya. Bila definisi “Selesai” untuk *Increment* adalah bagian dari sebuah ketentuan, standar, atau panduan pengembangan produk di dalam organisasi, maka semua *Scrum Team* di dalam organisasi harus mengikutinya sebagai kebutuhan minimal.

Bila “Selesai” untuk *Increment* bukan merupakan bagian dari sebuah ketentuan di dalam organisasi pengembangan produk, maka *Development Team* dari *Scrum Team* harus membuat definisi “Selesai” yang sesuai dengan kebutuhan produk. Apabila ada beberapa *Scrum Team* yang mengembangkan satu sistem atau produk secara bersamaan, masing-masing *Development Team* dari semua *Scrum Team* harus saling menyepakati definisi “Selesai” ini.

Setiap *Increment* merupakan tambahan untuk *Increment* sebelumnya dan sudah dipastikan telah diuji secara saksama guna memastikan seluruh *Increment* berfungsi secara satu kesatuan utuh.

Seiring bertambah dewasanya *Scrum Team*, diharapkan definisi “Selesai” mereka akan berkembang sehingga mencakup kriteria yang lebih ketat untuk kualitas yang lebih tinggi. Ketika diterapkan, definisi-definisi baru ini akan menambah pekerjaan baru yang perlu diselesaikan kembali terhadap potongan-*Increment* yang telah “Selesai” sebelumnya. Produk atau sistem apapun harus memiliki definisi “Selesai” sebagai standar untuk pekerjaan yang akan dilakukan.

Catatan Penutup

Scrum bersifat gratis dan dijabarkan di dalam panduan ini. Peran-peran, acara-acara, artefak-artefak, dan aturan-aturan Scrum tidak dapat diubah dan walaupun menerapkan hanya sebagian dari Scrum saja memungkinkan untuk dilakukan, namanya tidak bisa disebut Scrum. Scrum hanya ada bila diterapkan secara keseluruhan dan berfungsi dengan baik sebagai wadah untuk teknik, metodologi, dan praktik lain.

Ucapan Terima Kasih

Pihak terkait

Dari sekian banyak orang yang berkontribusi terhadap Scrum, kami harus memilih beberapa orang yang sangat berperan sejak awal : Jeff Sutherland bersama Jeff McKenna dan John Scumniotales, dan Ken Schwaber bersama Mike Smith dan Chris Martin, mereka semua bekerja bersama-sama. Banyak orang lain yang berkontribusi dalam tahun-tahun berikutnya dan tanpa bantuan dari mereka, Scrum tidak akan bisa disempurnakan seperti sekarang.

Sejarah

Ken Schwaber dan Jeff Sutherland merumuskan Scrum hingga tahun 1995 dan mempresentasikannya di konferensi OOPSLA pada tahun 1995. Presentasi ini pada dasarnya berisi hasil pembelajaran yang didapatkan oleh Ken dan Jeff dari beberapa tahun sebelumnya dan merupakan definisi formal awal mengenai Scrum.

Sejarah Scrum dijelaskan di tempat lain di luar panduan ini. Kami memberikan penghargaan untuk perusahaan-perusahaan yang mencoba Scrum pertama kali dan menyempurnakannya, diantaranya adalah Individual Inc, Newpage, Fidelity Investments, dan IDX (sekarang bernama GE Medical).

Panduan Scrum memuat Scrum sebagaimana dikembangkan dan dipertahankan selama 20 tahun lebih oleh Jeff Sutherland dan Ken Schwaber. Pihak-pihak lain menyediakan pola, proses dan masukan yang dapat digunakan dengan kerangka kerja Scrum. Hal ini dapat meningkatkan produktivitas, nilai bisnis, kreativitas dan tingkat kepuasan terhadap hasilnya.

Penghargaan untuk penerjemah

Panduan ini telah diterjemahkan dari bahasa aslinya versi bahasa Inggris yang telah disediakan oleh pengembang yang disebut di atas. Tim inti dari terjemahan Bahasa Indonesia ini adalah: Rudy Rahadian, Dayu Bagus, Joshua Partogi, Aditya Suryomurtjito, Asyirini Fajrin, Artanto Ishaam, Jiorutte Banjarnahor, Fenny Valentina, Ferdinan Wirawan, dan Edo Surya Pamungkas.

Mereka semua bertanggung gugat terhadap terjemahan Bahasa Indonesia Panduan Scrum.

Perubahan dari versi 2016 ke versi 2017

1. Penambahan bagian Penggunaan Scrum

Scrum dikembangkan untuk mengelola dan mengembangkan produk. Scrum mulai digunakan sejak dan telah digunakan secara meluas di seluruh dunia, untuk:

1. Meneliti dan menggali potensi pasar, teknologi dan kemampuan produk;
2. Mengembangkan produk dan peningkatan-peningkatannya;
3. Merilis produk dan peningkatan-peningkatannya, sesering mungkin di setiap hari;
4. Mengembangkan dan memelihara operasional sistem komputasi awan (daring, keamanan, sesuai permintaan) dan lingkungan operasional lain untuk penggunaan produk; dan,
5. Mengelola dan memperbarui sebuah produk.

Scrum telah digunakan untuk mengembangkan perangkat lunak, perangkat keras, perangkat lunak terintegrasi, aplikasi dalam jaringan yang saling berinteraksi, kendaraan tanpa awak, sekolah, pemerintahan, pemasaran, mengelola operasional organisasi dan hampir semua hal yang kita gunakan di kehidupan sehari-hari sebagai seorang individu dan anggota masyarakat.

Keampuan Scrum dalam menghadapi kompleksitas semakin terbukti setiap harinya seiring dengan semakin meningkatnya kompleksitas dan interaksi antara teknologi, pasar dan lingkungan.

Scrum terbukti efektif dalam transfer pengetahuan secara berkala dan berkelanjutan. Scrum saat ini sudah digunakan secara luas untuk produk-produk, layanan-layanan, dan manajemen perusahaan induk.

Esensi dari Scrum adalah sebuah tim kecil yang terdiri dari beberapa orang. Tim ini bersifat sangat fleksibel dan mampu beradaptasi. Kekuatan ini terus berlanjut dalam satu tim, beberapa tim, banyak tim, maupun banyak tim yang berhubungan dalam mengembangkan, merilis, mengoperasikan dan menjaga pekerjaan; dan produk hasil pekerjaan dari ribuan orang. Mereka berkolaborasi dan saling berinteraksi melalui arsitektur pengembangan dan target lingkungan rilis produk yang mutakhir.

Pada saat kata “mengembangkan” dan “pengembangan” digunakan di dalam Panduan Scrum ini, keduanya mengacu pada pekerjaan yang kompleks seperti tipe pekerjaan yang telah dipaparkan di atas.

2. Perubahan penjelasan di bagian *Scrum Master* untuk memberikan kejelasan mengenai peran ini. Bagian ini sekarang menjadi:

Scrum Master bertanggung jawab untuk mengenalkan dan menyokong penggunaan Scrum sebagaimana dijelaskan di dalam Panduan Scrum ini. *Scrum Master* melakukan ini dengan membantu orang-orang agar dapat memahami teori, praktik-praktik, aturan-aturan dan tata nilai Scrum.

Scrum Master adalah pemimpin yang melayani *Scrum Team*. *Scrum Master* membantu orang-orang di luar *Scrum Team* untuk dapat memahami interaksi mana yang bermanfaat dan tidak bermanfaat. *Scrum Master* membantu orang-orang untuk mengubah interaksi ini guna memaksimalkan nilai bisnis yang dihasilkan oleh *Scrum Team*.

3. Penambahan di bagian Pelayanan *Scrum Master* kepada *Product Owner*

Memastikan tujuan, ruang lingkup dan ranah produk dipahami sebaik mungkin oleh semua orang di dalam *Scrum Team*;

4. Perubahan di paragraf pertama *Daily Scrum* yang sekarang menjadi:

Daily Scrum adalah acara untuk *Development Team* yang memiliki batasan waktu 15 menit. Acara ini dilakukan setiap hari selama *Sprint* berlangsung. Di acara ini, *Development Team* membuat rencana kerja untuk 24 jam ke depan. Acara ini mengoptimalkan kolaborasi dan performa dari tim dengan melakukan inspeksi pada pekerjaan yang dilakukan semenjak *Daily Scrum* sebelumnya dan melakukan prakiraan terhadap pekerjaan selanjutnya di dalam *Sprint*. *Daily Scrum* dilakukan di waktu dan tempat yang sama setiap harinya untuk mengurangi kompleksitas.

5. Perubahan di bagian *Daily Scrum* untuk memberikan kejelasan mengenai tujuan dari *Daily Scrum* yang sekarang mengandung tulisan ini:

Struktur dari pertemuan ini ditentukan oleh *Development Team* dan bisa diadakan lewat berbagai macam cara selama pertemuan ini fokus terhadap kemajuan menuju *Sprint Goal*. Beberapa *Development Team* akan menggunakan pertanyaan-pertanyaan, beberapa akan lebih berdiskusi. Berikut adalah contoh pertanyaan yang mungkin saja digunakan:

- Apa yang telah saya lakukan kemarin untuk membantu *Development Team* mencapai *Sprint Goal*?
- Apa yang akan saya lakukan hari ini untuk membantu *Development Team* mencapai *Sprint Goal*?
- Apakah saya melihat ada hambatan yang menghalangi saya ataupun *Development Team* dalam mencapai *Sprint Goal*?

6. Menambahkan penjelasan mengenai batasan waktu

Penggunaan kata “paling lama” untuk menghindari adanya pertanyaan jika acara-acara *Scrum* harus memiliki durasi sesuai ketentuan melainkan waktu maksimal yang telah dialokasikan.

7. Penambahan di bagian *Sprint Backlog*

Untuk memastikan adanya peningkatan berkelanjutan, *Sprint Backlog* berisi setidaknya satu peningkatan proses dengan prioritas tertinggi dari hasil pertemuan *Retrospective Sprint* sebelumnya.

8. Menambahkan penjelasan di bagian *Increment*

Sebuah *Increment* adalah hasil pekerjaan yang bisa diinspeksi dan telah selesai guna mendukung empirisme di akhir *Sprint*. *Increment* adalah sebuah langkah kecil menuju sebuah visi ataupun tujuan.